

//Alumno: Tarazona Silva Cristian **Código:** 18190319

1. Describir como mínimo tres paradigmas de programación y dar sus características

PARADIGMA DE PROGRAMACIÓN IMPERATIVA

- Paradigma de programación estructurada:
 - ✓ Sólo utiliza secuencia, decisión y repetición
 - ✓ Código más fácil de leer, pero todavía difícil para sistemas grandes debido a la repetición de código
- Paradigma de programación secuencial
 - ✓ Sinónimo: paradigma procedural
 - ✓ Uso de sub-programación
 - Agrupación de código permitiendo la creación de acciones complejas
 - Asignación de un nombre para esas acciones complejas
 - Llamada a esas acciones complejas desde cualquier punto del programa
 - ✓ Estas acciones complejas se denominan procedimientos, subrutinas y funciones
- Paradigma de programación orientado a objetos
 - ✓ Agrupación de procedimientos y variables
 - Paquetes de clases
 - ✓ Agrupamiento de clases
 - ✓ Clases de objetos
 - ✓ Representan bibliotecas de apoyo

PARADIGMA DE PROGRAMACIÓN DECLARATIVA

- Paradigma de programación lógica
 - ✓ Unificación de términos
 - ✓ Mecanismos de inferencia automática
 - ✓ Recursión como estructura de control básica
 - ✓ Visión lógica de la computación
- Paradigma funcional
 - ✓ Programación declarativa
 - ✓ Definición y evaluación de funciones
 - ✓ Uso de la recursión
 - ✓ Funciones como datos primitivos
- Base de datos
 - ✓ Modelo Externo (visiones de usuario)
 - ✓ Modelo “Lógico”
 - ✓ Modelo Interno (nivel físico)

2. ¿Cuál es la diferencia entre `i++` e `++i`, dar un ejemplo en código?

Pre-incremento

`++x`, El valor de la expresión es el valor de la variable después que el nuevo valor es almacenado, ejemplo.

```
int x = 7;
```

```
System.out.println(++x); // imprime 8.
```

```
System.out.println(x); // imprime 8.
```

Post-incremento

`x++`, El valor de la expresión es el valor de la variable antes que el nuevo valor sea almacenado, ejemplo.

```
int x = 7;
```

```
System.out.println(x++); // imprime 7
```

```
System.out.println(x); // imprime 8.
```

3. Completar el cuadro con la información de prioridad siendo uno más prioritario que 5.

Operador	Prioridad	Operador	Prioridad
/	4	(expr)	1
--var	3	+expr	3
*	4	+	5
%	4	-	5
var--	2	&&	5

4. Responder las siguientes preguntas y dar ejemplo.

a) ¿Qué significa un casting en programación?

Es un procedimiento para transformar una variable primitiva de un tipo a otro. También se utiliza para transformar un objeto de una clase a otra clase siempre y cuando haya una relación de herencia entre ambas.

```
int edad=25;
int edad2=17;
int edad3=23;
double promedio;
promedio = (double)(edad+edad2+edad3)/3;
System.out.println(promedio);
```

Imprimirá: 21.666666666666668 ;

b) ¿Qué es una función y un procedimiento?

Funciones: permiten automatizar tareas que requerimos con frecuencia y que además se pueden generalizar por medio de parámetros o argumentos.

Las funciones son un conjunto de líneas de código (instrucciones), encapsulados en un bloque, usualmente reciben parámetros, cuyos valores utilizan para efectuar operaciones y adicionalmente retornan un valor. En otras palabras una función puede recibir parámetros o argumentos (algunas no reciben nada), hace uso de dichos valores recibidos como sea necesario y retorna un valor usando la instrucción `return`, si no retorna algo, entonces no es una función. En java las funciones usan el modificador *static*, ejemplo.

```
int metodoEntero()//Función sin parámetros
{
    int suma = 5+5;

    return suma; //termina la ejecución del método

    //return 5+5; //Este return nunca se ejecutará

    //Intenta intercambiar la línea 3 con la 5

    //int x = 10; //Esta línea nunca se ejecutará
}
```

Procedimiento: Los procedimientos son básicamente un conjunto de instrucciones que se ejecutan sin retornar ningún valor, hay quienes dicen que un procedimiento no recibe valores o argumentos, sin embargo en la definición no hay nada que se lo impida. En el contexto de Java un procedimiento es básicamente un método cuyo tipo de retorno es *void* que no nos obliga a utilizar una sentencia `return`, ejemplo.

```
void procedimiento(int n, String nombre) //Notar el void
{
    if (n > 0 && !nombre.equals(""))//dos parámetros
    {
        System.out.println("hola " + nombre);

        return; //Si no ponemos este return se mostraría hola y luego adiós
    }

    System.out.println("adios");
}
```

c) ¿Qué quiere decir sobrecarga de operadores?

Los operadores son un tipo de tokens que indican al compilador la realización de determinadas operaciones sobre variables u otros. La sobrecarga de operadores permite redefinir ciertos operadores, como '+' y '-', para usarlos con las clases que hemos definido. Se llama sobrecarga de operadores cuando reutilizando el mismo operador con un número de usos diferentes, y el compilador decide cómo usar ese operador dependiendo sobre qué opera.

7. Piden calcular el valor de X e i; del código dado.

X: 69 y i: 68